Large-scale, Fast and Accurate Shot Boundary Detection through Convolutional Neural Networks

Ahmed Hassanien*, Mohamed Elgharib*, Ahmed Selim†,
Sung-Ho Bae‡, Mohamed Hefeeda§ and Wojciech Matusik‡
*HBKU Qatar Computing Research Institute,
†Trinity College Dublin, CONNECT Center
‡MIT CSAIL
§Simon Fraser University

***Abstract*—Shot boundary detection (SBD) is an important pre-processing step for video manipulation. Here, each segment of frames is classified as either sharp, gradual or no transition. Current SBD techniques analyze hand-crafted features and attempt to optimize both detection accuracy and processing speed. However, the heavy computations of optical flow prevents this from happening. To achieve this aim, we present an SBD technique based on spatio-temporal Convolutional Neural Networks (CNN). Since current datasets are not large enough to train an accurate SBD CNN, we are the first to present a very large SBD dataset that allows deep neural networks techniques to be effectively applied. Our dataset contains more than 3.5 million frames of sharp and gradual transitions. The transitions are generated synthetically using image compositing models. Our dataset contains additional 70,000 frames of important hard-negative no transitions. We perform the largest evaluation to date for one SBD algorithm, on real and synthetic data, containing more than 4.7 million frames. In comparison to the state of the art, we outperform dissolve gradual detection, generate competitive performance for sharp detections and produce significant improvement in wipes. In addition, we are up to 11 times faster than the state of the art.**

I. Introduction

Videos are composed of different camera shots placed after each other. Shot transitions can be classified into two main categories: sharp and gradual [1] (see Fig. 1). Gradual transitions are classified into dissolve and non-dissolve. Disolve transitions include cases such as semi-transparent graduals, fade in and fade out (see Fig. 1). Non-dissolve are dominated by wipes (see Fig. 1). Wipe graduals have much wider variety than the dissolve graduals, which makes their detection harder. This led to very limited research for their explicit detection [2].

Current SBD techniques analyze hand-crafted features [3], [4], [1], [5], [6], [7], [8], [9], [10], [11]. Fast techniques analyze only spatial information such as intensity histogram [4], [10], edges [6], mutual information and others [7], [11], [9], [8]. Such techniques, while being fast, generate poor detection. To boost detection, motion information is incorporated through optical flow [3], [1], [12], [2]. However, the heavy computations of optical flow [13], [14], [15] make such techniques slow. Since SBD techniques are commonly used as a pre-processing step for video manipulation, optimizing both their detection accuracy and processing speed are important. This, however, remains a challenging problem.

We present DeepSBD, a fast and accurate shot boundary detection through convolutional neural networks (CNN). We exploit a big data set to achieve high detection performance. Our technique takes segments of 16 frames as input, and classifies them as either gradual, sharp, or no-transition through a 3D CNN architecture, inspired by C3D [16]. To train our network, we need a well annotated large dataset. Despite some datasets already exist from the TRECVID challenge and others [1], [17], experiments show they are not sufficient to train a high accuracy CNN solution. In addition, the vast majority of these datasets are used for testing and evaluating different techniques, and not for training. Instead, we present a large SBD dataset with clean and accurate annotations. This allows us to test on all available TRECVID data (3.9 million frames) [1]. The first dataset portion, SBD_Syn, is generated synthetically using image compositing models. It contains 220,339 sharp and gradual segments, each segment contains 16 frames. The second portion, SBD_HN, contains 4,427 no transition segments. They are carefully manually annotated in a way to improve detector's precision; they act as hard-negatives.

Aspects of novelty of our work include: 1) The first CNN SBD technique. We outperform dissolve gradual detection, generate competitive performance for sharp detections and produce significant improvement for wipes. In addition, we are up to 11 times faster than the state of the art. 2) A large SBD dataset for training an accurate CNN model. The dataset contains 3.5 million frames of synthetic transitions and 70,000 frames of hard negative no-transitions. 3) A large wipes dataset containing 1.1 million frames. We will release all our data-sets and code to encourage future research. 4) The largest SBD evaluation to date on 4.7 million frames. 3.9 million frames are from all TRECVID years [1] while the rest are synthetically generated.

II. State of the Art

Current SBD techniques are classified into two main categories: spatial-only and spatio-temporal analysis based. The former estimates the temporal profile by comparing only spatial features [3], [4], [1], [5], [6], [7], [8], [9], [10], [11]. A number of spatial features are used such as color histograms [4], [10], edges [6], mutual information and Entropy [7],

wavelet representations [3], SURF [18] and many others [11], [9], [8], [19]. Spatial-only SBD techniques generate conservative detection accuracy with fast processing speed. Spatio-temporal techniques use optical flow to make detection more robust to scene and camera motions [3], [5], [20], [21], [21]. Such motions can arise due to camera movements and shakiness and often confuses the detection process. Hence, optical flow [13], [14], [15] between neighboring frames is estimated and removed through frame interpolation. Analysis of the temporal profile is then proceeded as in the spatial-only techniques. Here, motion compensation often reduces false detections of SBD. The main drawback of spatio-temporal techniques, however, is the heavy computations of optical flow.

Among the rich SBD literature, four of the best performing and/or most recent techniques are Liu et al. [2], Yuan et al. [22], Lu et al. [4] and Priya et al. [3]. Lu et al. focuses more on generating fast results and hence they do not incorporate motion information. Their technique is based on assessing temporal discontinuities through HSV histogram. Priya et al. [3] proposed a wavelet based feature vector that measures four main quantities: color, edge, texture, and motion strength. To the best of our knowledge, Liu et al. [2] is the latest wipe detector. A candidate transition segment is proposed and the difference between each frame and the start and end frame is calculated. This generates two curves, one for the start and another for the end of the segment. For wipes, the curves should have opposing gradients and somewhat linear. Furthermore, to reduce errors due to camera and object movements, motion compensated frame differencing is used.

SBD Datasets Between the years 2001 to 2007, the National Institute of Standards and Technology (NIST) [23] maintained data for the TRECVID shot boundary detection (SBD) challenge [1]. The dataset contains a wide variety of contents including color, gray-scale, indoor, outdoor, outer-space and different levels of noise. The dataset has a total of 4,333,153 frames with 24,423 transitions, 64% of which are sharp. The rest are gradual. Transitions were manually annotated in a way to distinguish between sharp and graduals. Four more releases from a different challenge were maintained by NIST that contain data relevant to SBD. The releases are T2007t, T2007d, T2008 and T2009, containing 34,765,424 frames with 155,902 transitions. The annotations of these data, however, do not distinguish between sharp and graduals. Finally, one more data release related to SBD was generated by Baraldi et al. [17]. Here, the authors addressed the different application of video scene segmentation.

III. Our Approach

A. Algorithm Design

We present a technique for automatic detection and classification of shot boundaries. We name our technique DeepSBD. A video is divided into segment of frames. Each segment is assigned one of three labels: 1) sharp transition, 2) gradual transition or 3) no transition. We use segments of length 16, with an overlap of 8. Each segment is fed to a deep 3D-CNN that analyzes both spatial and temporal information. Our network, C3D_sbd, is inspired by [16] and is trained from scratch for shot boundary detection. The last layer is followed by softmax for classification, this gives the initial labeling estimate. Consecutive segments with the same labeling are merged and the result is passed to a post-processing step. The step removes false positives containing little motion. For such segments, we estimate the color histogram of the first and end frame. We measure the Bhattacharyya distance between these histograms. If the distance is small, we declare this segment as no-transition. We use an OpenCV implementation for both color histogram and Bhattacharyya distance, which is very fast.

Fig. 1: Shot transitions are classified into two main categories: sharp and gradual. Gradual transitions are further classified into soft and wipes. Soft include semi-transparent, fade in and fade out.

Our network, C3D_sbd, consists of five 3D convoluational layers. All convolutional layers are followed by Rectified Linear Unit (ReLU) and pooling layers. The first two convolutional layers are followed by Local Response Normalization (LRN). Two fully connected layers exist, fc6 and fc7, each containing 2049 neurons. The last fully connected layer fc8 contain only 3 neurons, one for each class (sharp, gradual and no transition) followed by softmax. In comparison to [16], C3D_sbd uses batch normalization after the first two convoluational layers.

B. Dataset Generation

Training an SBD CNN requires a large and well annotated dataset. We present two datasets, SBD_Syn (Table I) and SBD_HN (Table II). SBD_Syn is generated synthetically while SBD_HN is generated in a way to improve detector's precision, through bootstrapping. We first use SBD_Syn with T2005 and Baraldi et al. to train from scratch our solution. We run this solution on data from T2007t/d, T2008 and T2009 in order to label each segment to sharp, gradual, or no transition. Due to the massive size of these datasets, however, we only examine surroundings segments that originally annotated as any form of transition. Note that original annotations here do not distinguish between sharp or gradual. We closely examine segments detected as graduals. We manually filter

them into three classes: gradual, sharp and no transitions. The no-transition represents complicated hard-negative cases such as illumination variation and fast motion (see Figure 2). We call this dataset portion SBD_HN. Finally, we train from scratch a final solution using both SBD_Syn and SBD_HN. We optionally use T2005 and Baraldi et al. to further improve performance. Results show that SBD_HN has a great impact in reducing false detections and improving the overall performance. The supplementary material shows images from the datasets of SBD_Syn and SBD_HN in Figure 1 and Figure 2

SBD_Syn: Table I shows the content of SBD_Syn. The dataset is generated synthetically through image composting models [24]. A transition is modeled as a linear combination between the underlying shots

$$I_t(\mathbf{x}) = \alpha_t(\mathbf{x})B_t(\mathbf{x}) + (1 - \alpha_t(\mathbf{x}))F_t(\mathbf{x}) \quad (1)$$

Here, I_t denotes the observed frame at time t, while B and F are the contents from the previous and next shots respectively. α is the mixing parameter between both shots while $\mathbf{x}$ denote image pixels. The values and distribution of α define the type of shot transition. If no transition exists, then $(\alpha_t, \alpha_{t+1}) = (1, 1)$. Sharp transitions, however, have a sudden temporal change with $(\alpha_t, \alpha_{t+1}) = (1, 0)$. For gradual transitions, α change over time from 1 to 0. This change occurs over a set of frames and hence $(\alpha_t, ...\alpha_t + N) = (1, ., ., 1 - t/N, ., 0)$. N is the transition duration and t is the frame index where $t = 0$ denotes the last frame of the previous shot. Here, the in-between α values are non-binary. This generates the dissolve nature of most gradual transitions (Figure 1). For wipes, α is varying spatially aswell as temporally.

To generate SBD_Syn we need to define F, B and α of Eq. 1. F and B must not contain any shot transitions. We use the T2007t/d, 2008, 2009 and their annotations to find such frames. We sample F and B in a way to ensure a large offset from the nearest transition. Sharp transitions are generated by applying Eq. 1 with $(\alpha_t, \alpha_{t+1}) = (1, 0)$. Gradual transition generation, however, is more complex. For SBD_Syn we focus on dissolve gradual generation. We randomly select the transition duration N, where $N <= 16$. We also randomly select the transition start and end frame for B and F. We draw N α samples, where α is modeled with a uniform distribution. We sort all α values in descending order and apply Eq. 1 for each of the N frames.

We train C3D_sbd using balanced data for sharp, gradual and no-transition. We experimented with different data sizes. We found 40,000 segments for each class generates good results. We use a step learning policy. Learning rate starts with a value of 0.0001 and is reduced gradually by a factor of 10 every two epochs. We use a batch size of 20, and train the model for 6 epochs. That is two epochs for each learning rate of 1e-4, 1e-5, and 1e-6. The momentum value is 0.9. All these values were set empirically to optimize performance.

IV. Results

We performed experiments on real data as well as on synthetically generated data. We examined 4,683,552 frames, 81.8% of which are real. Our work presents the largest SBD evaluation to date for one algorithm. We asses performance quantitatively using precision (P), recall (R) and F-score (F). Here, we use the standard TRECVID evaluation metric [1] where a transition is detected if it overlaps with the annotations by at least one frame. We report the per-transition performance. During comparison we highlight the best performing technique in **bold**. In order to account for possible mis-annotations and system errors in such large experiments, we claim a technique is superior only if it achieves more then 0.5% P, R, or F improvement over others. Techniques within 0.5% difference are claimed as competitive. We train two models, both using our datasets DSB_Syn and SBD_HN. One of them uses a few real data from T2005 and Barladi et al. , denoted by r, at most 7% of the total training data. Both models are competitive to each other. More detailed results are reported in the supplementary material.

We compare against the latest techniques (Lu et al. [4], Priya et al. [3], Apostolidis et al. [18] and Berladi et al. [19]) as well as the best performers in the 7 years of the

TABLE I: Our dataset SBD_Syn in terms of number of segments (16 frames each). SBD_Syn is synthetically generated from T2007t/d,2008,2009 using image compositing. The data contains a balanced portion of no-transitions.

Datasets	Synthetic Gradual	Synthetic Real
T2007t	19398	19439
T2007d	13656	19607
T2008	39047	32456
T2009	44158	32578
Total	116259	104080

TABLE II: Our SBD_HN dataset. The no-transition represent complex hard-negatives (Figure 2). When included in training, precision is significantly is improved.

Transitions	Number of segments
No-transition	4,427
Gradual	11,249
Sharp	359
Total	16,035

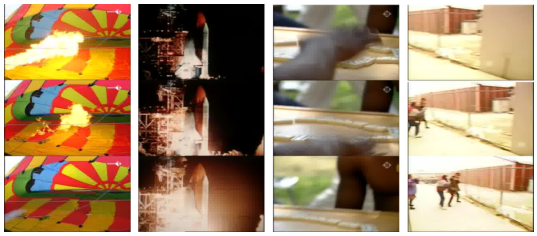

Fig. 2: No-transition samples from our hard negative data (SBD_HN). They contain complex cases such as fast motion, occlusion and illumination variation. These cases can be misclassified as graduals. More examples are in the supplementary material, Figure 2

TABLE III: Videos of T2001a and T2001b

	videos names
T2001a	BOR10_001, BOR10_002, NAD57, NAD58, anni001, anni005, anni006, anni007, anni00, anni009, anni010
T2001b	BOR03, BOR08, BOR10, BOR12, BOR17

TRECVID challenge (Yuan et al. [12] and Liu et al. [2]). These techniques show the compromise between detection accuracy and processing speed commonly present in SBD. Lu et al. [4] is the fastest of all, but generates conservative performance. Priya et al. [3], Liu et al. [2] and Yuan et al. [12] generate better performance. However, at the cost of heavy optical flow computation. Our results show that DeepSBD optimizes both detection accuracy and processing speed over all current techniques. That is, we outperform gradual detection, generate competitive performance for sharp transitions and produce significant improvement in wipes detection. In addition, we are up to 11 times faster than the state of the art. More detailed results are reported in the supplementary material.

A. Real Sequences

We evaluated our technique on all seven TRECVID releases, from 2001 to 2007. They have a total of 3,831,648 frames, with 8,545 gradual and 14,602 sharp transitions. No test data was included in the training. Table IV shows performance evaluation on 6 sequences commonly used in Lu et al. [4] and Priya et al. [3]. The sequences are from T2001a (see Table III) and present challenging videos from outer-space. The videos include cases such as global illumination variation, smoke, fire and fast non-rigid motion. We outperform Lu et al. in all sequences for both gradual and sharp transition. Furthermore, we outperform Priya et al. in the vast majority of sequences in both transition types.

Table V compares our technique against Priya et al. [3] on T2007. Note that [3] used a slightly different approach for evaluation than the one recommended by TRECVID [1]. TRECVID recommends estimating the average performance per transition. However, [3] estimated the average performance per sequence. Furthermore, Priya et al. tested on 17 sequences, 7 of which were included in their training set. This biases the results towards Priya et al. [3]. Hence, for fair comparison these 7 sequences should be removed from the 17 test sequences and the comparison should be done on at most 10 sequences. To illustrate this point, we examined our technique with different sizes of the test dataset. Each column of Table V shows the performance with different size of the test data. With 10 test sequences, our technique outperforms Priya et al. [16] significantly in gradual transitions (0.88 vs. 0.76 f-score) and generates competitive results for sharp transitions. Furthermore, we still outperform Priya et al. even with a test-set of 14 sequences. Here, however, at least 4 sequences are included in Priya et al. training and hence results are biased towards Priya et al. Including these videos in our training is expected to improve performance even further. The spatio-temporal aspect of our solution allow us to generate these high detection accuracy results without explicitly estimating optical flow. Experimentation showed that just relying on the spatial information generates very poor results.

Table VI evaluates DeepSBD on T2004, 2005, 2006 and 2007. To test on 2005, we removed it from our training. We compare against the best TRECVID performers as well as Lu et al. [4]. We significantly outperform Lu et al. in T2007. Furthermore, we outperform the best TRECVID performers, Liu et al. [2] and Yuan et al. [12] on all four datasets. Table VII evaluates DeepSBD on the remaining TRECVID datasets. T2001b and 2002 annotations contain significant overlap between sharp and gradual transitions. Hence, for them we show the overall combined transitions performance. Furthermore, T2003 is missing 4 videos and hence we could not compare against the reported TRECVID performance. In all sequences we generate good performance. T2001b and 2002 sequences contain strong noise and jitter. Yet, our technique was robust enough to handle such artifacts. Figure 3 (a) shows the precision-recall curves for our DeepSBD on all real TRECVID sequences. The supplementary material (Tables II-XV) shows the per sequence results for each of the TRECVID dataset examined by our technique and includes much more statistics e.g. true positives (TP), false positives (FP), false negative (FN) and so on. We also evaluated our technique on the RAI RAI dataset [25]. We generated an f-score of 0.94, which compares favorably with the score of 0.84 for both Apostolidis et al. [18] and Berladi et al. [19].

B. The importance of our datasets

Table VIII shows the significance and importance of our datasets SBD_Syn and SBD_HN in generating high accuracy detections. We evaluate DeepSBD on T2007 with six different training sets: 1) R_3-5 2) R_3-6 3) R_3-6 + HN, 4) S + r, 5) S + r + HN and 6) and S + HN. S and HN is short for our datasets SBD_Syn and SBD_HN. R_3-6 represent TRECVID real videos and annotations from 2003 to 2006. r is T2005 and Baraldi [17]. Results show that training with R_3-5 generates poor performance. In addition, it limits us to testing on just 3 data-sets (T2001a, T2006 and T2007). Adding T2006 to training improves performance but limits our testing further to 2 data-sets (T2001a and T2007). Adding our hard negative data SBD_HN (HN) improves precision and performance significantly. This shows the high quality and importance of our SBD_HN. The best performance, however, is generated when both our datasets SBD_Syn and SBD_HN with r are used for training. In addition to the highest performance, this option allows us to test on all TRECVID videos, except T2005. Removing r from the training generates the second best performance. This, however, allow us to test on all TRECVID videos, including T2005. The experiment shows the significance and importance of our data-sets. We performed this experiment on several test sets and we found S + r + HN and S + HN are always the top and competitive to each other (see supplementary material, Table. 1). This shows

TABLE IV: DeepSBD evaluation on 6 challenging sequences from TRECVID 2001 (D1-D6). Our technique outperforms Lu et al. and Priya et al. [3] in the vast majority of sequences. The improvement is more significant in gradual transitions.

	D1-anni5	D2-anni6	D3-anni9	D4-anni10	D5-NAD57	D6-NAD58
Lu et al. [4]						
Abrupt	-	0.905	0.754	0.892	-	0.962
Gradual	-	0.817	0.824	0.734	-	0.884
Priya et al. [3]						
Abrupt	**0.85**	0.911	0.842	0.897	0.945	**0.945**
Gradual	0.938	0.885	0.873	0.822	0.809	0.885
DeepSBD (ours)						
Abrupt	0.818	**0.988**	**0.961**	**0.918**	**0.957**	0.904
Gradual	**0.945**	0.885	**0.919**	**0.855**	**0.917**	**0.914**

TABLE V: Per-sequence f-score on T2007. We compare against Priya et al. [3] on test-sets of different sizes. Since Priya et al. [3] is trained on 7 out of the total 17 test sequences, comparison should be done on at most 10 sequences. Results show we significantly outperform Priya et al. with a test-set of 10 sequences. Here, our gradual transitions detector is more than 12% better than Priya et al. in f-score, a significant improvement due to our CNN solution. Furthermore, we still outperform Priya et al. with a test-set size up to 14 sequences. Here, however, at least 4 sequences were included in Priya et al. which biases the results towards Priya et al. Including these videos in our training is expected to boost our performance even further.

Size of test-data (in sequences)	9	10	11	12	13	14	15	16	17
Priya et al. [3]									
Abrupt	0.9733	0.974	0.9748	0.9742	0.9737	0.9741	0.9733	0.9737	0.974
Gradual	0.775	0.7578	0.7677	0.7742	0.7811	0.7825	0.7802	**0.7726**	**0.78**
DeepSBD (ours)									
Abrupt	0.9729	0.9749	0.9743	0.974	0.9743	0.9749	0.9733	0.9713	0.9726
Gradual	**0.8962**	**0.8774**	**0.8613**	**0.8395**	**0.8171**	**0.797**	0.7758	0.7507	0.7259

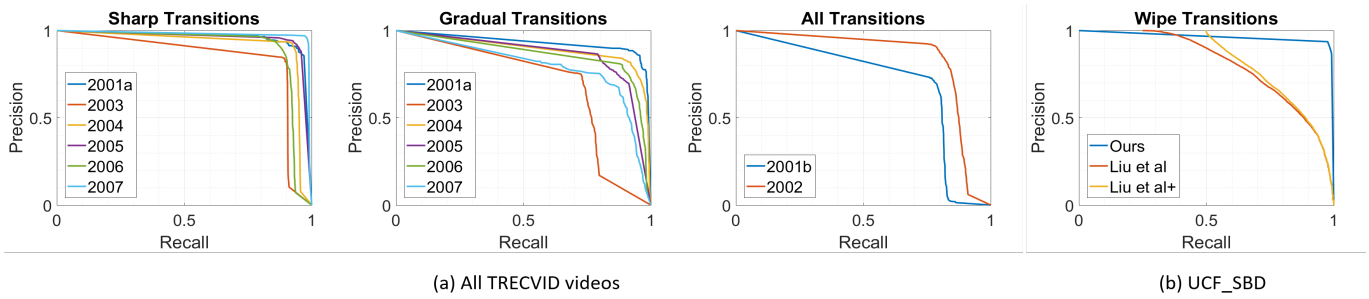

Fig. 3: (a) Precision-Recall of DeepSBD for all TRECVID sequences. (b) Precision-Recall of DeepSBD for wipes. Here, we compare against two implementations of Liu et al. [2] wipe detector. The first implementation examines all frames of UCF101_SBD and '+' examines only frames not classified as sharp nor gradual by DeepSBD. We significantly outperforms both approaches.

the significance of our datasets and their generation process (Section III-B).

C. Controlled Experiments

We generated a synthetic test-set. Our dataset contain 53,324 segments, divided equally between gradual, sharp, wipes and no-transitions. Each segment is 16 frames long. We generated gradual and sharp transition using image compositing as we did for SDB_Syn (see Eq. 1). Here, we constrain the shots to come from two different UCF101 videos [26]. We present the first large wipes dataset, containing 1.1 million wipe frames (20% test). They are also generated using Eq. 1. Here, however, the opacity values α have more complicated spatio-temporal patterns than sharp and gradual transitions. Figure 5 shows some of the 196 α mattes we used. The supplementary material, Figure 3, shows frames from our wipes dataset. We call our synthetic UCF dataset UCF101_SBD.

We train the model using SBD_Syn, SBD_HN and the synthetic wipes. This model generates 4 classes. Table IX evaluates DeepSBD on UCF101_SBD. We generate high performance for all classes, including wipes. Performance is

TABLE VI: Comparing DeepSBD against different techniques. We compare against TRECVID best performers, Liu et al. [2] for 2006/2007 and against Yuan et al. [22] for 2004/2005. We also compare against the latest no optical flow technique of Lu et al. [4]. We outperform all techniques on all datasets.

	T2004	T2005	T2006	T2007
Best TRECVID performers [1]				
Abrupt	0.929	0.935	0.899	0.972
Gradual	0.806	0.786	0.814	0.753
Lu et al. [4]				
Abrupt	-	-	-	0.761
Gradual	-	-	-	0.618
DeepSBD (ours)				
Abrupt	0.926	0.934	0.895	0.973
Gradual	**0.866**	**0.844**	**0.895**	**0.776**

TABLE VII: Evaluating DeepSBD on T2001a, 2001b, 2002 and 2003.

	T2001a	T2001b	T2002	T2003
DeepSBD				
Abrupt	0.923	-	-	0.866
Gradual	0.906	-	-	0.759
Overall	0.915	0.748	0.865	0.8337

Fig. 4: Failure cases in gradual transition detection.

higher than the ones previously reported on the TRECVID sequences. This could be due to the highly accurate annotations of UCF101_SBD. Figure 3 (b) compares our wipe detector against the state of the art of Liu et al. [2]. We evaluate Liu et al. using two strategies. The first examines all frames of UCF101_SBD. The second, '+', examines only frames not detected as gradual nor sharp transitions by DeepSBD. Our technique outperform both approaches significantly.

D. Processing Speed

We examined a TRECVID video of duration 4,096 seconds containing 102,400 frames. We ran the test-phase of DeepSBD with batch size 100 segments. The GPU performs 64 iterations in order to process the 102,400 frames are processed. The smaller the batch size, the more iterations required and hence the more time required to process all frames. However, the less memory is required. Experiments shows that the processing speed gain from 10 to 100 batch size is between 16-19.3 real-time speed up factor.

We use Titan X, a GPU commonly used for deep learning applications. Table X compares the processing speed of different SBD techniques. In comparison with the best performing optical-flow based techniques, we are 11 times faster than

TABLE VIII: Training DeepSBD with different datasets. Results show that best performance is generated when our both SBD_Syn (S) and SBD_HN (HN) and r are used. Removing any real sequences (r) from our datasets generates the second best performance (S+HN). The advantage of this option is allowing us to test on all TRECVID videos, including T2005 (Table VI). The table also shows SBD_HN clearly improves the precision and overall performance.

	P	R	F	P	R	F
R_3-5	0.495	0.665	0.568	0.894	0.872	0.883
R_3-6	0.683	0.683	0.683	0.957	0.95	0.953
R_3-6 + HN	0.755	0.705	0.729	0.961	0.961	0.961
S + r	0.722	0.63	0.673	0.979	0.955	**0.967**
S + r + HN	**0.799**	**0.753**	**0.776**	**0.973**	**0.969**	**0.971**
S + HN	0.779	0.714	0.745	0.969	0.966	**0.968**

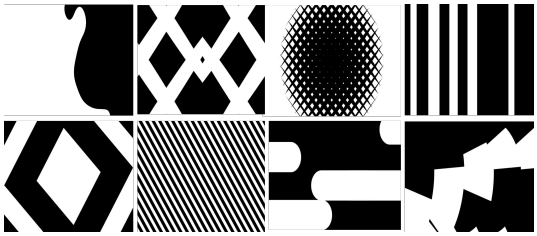

Fig. 5: Samples of the α mattes used for generating the wipes of UCF101_SBD.

TABLE IX: Evaluating DeepSBD on our synthetic dataset UCF101_SBD. In total UCF101_SBD has $53,253$ segments, divided equally among all classes (normal, gradual, sharp and wipes). Each segment has 16 frames. DeepSBD generates a very high performance in all classes.

Transition Type	Precision	Recall	F-measure
Gradual	0.989	0.995	0.992
Sharp	0.984	0.999	0.992
Wipes	0.976	0.936	0.956

TABLE X: Real-time speed-up factor for different SBD techniques. We are up to 11 times faster than all techniques.

	Real-time speed-up factor
DeepSBD	19.3
Liu et al. [2], [1]	3.24
Priya et al. [3]	1.76
Yuan et al. [22]	2.43

Priya et al. [3], 6 times faster than Liu et al. [2] and 9.65 times faster than Yuan et al. [22]. The supplementary material shows more analysis of the processing speed in Figure 4-5 (Section II)

E. What did the CNN learn?

We randomly selected two segments (16 frames) from UCF101 and synthetically generated a sharp and gradual transition using Eq. 1. We treated one of the two sequences as no-transition. We examined all segments using DeepSBD. Figure 6 shows the heat map of some Conv5 filter responses for each transition type. The filters are stacked next to each other, in blocks. The red grid shows some filters' borders. Time is the y-axis and space is the x-axis. Vertical space is averaged over the horizontal space. Sharp transitions have abrupt responses in the time axis in form of bright horizontal lines. Gradual transitions have blurred responses in the time axis. No transitions do not show a specific response pattern. The patterns are consistent on several other segments. The supplementary material shows more of such results in Figure 6 (Section III).

V. Conclusion

We presented the first CNN technique for shot boundary detection. Current techniques compromise between detection accuracy and processing speed and use hand-crafted features. We exploit big data to optimize both accuracy and speed. This is important as SBD is a common pre-processing step for video manipulation. We present two large datasets containing 3.57 million frames. One set is generated synthetically while the other is carefully annotated through bootstrapping. We outperform state of the art gradual transition detections, generate competitive performance in sharp transitions and produce significant improvement in wipes detections. Our approach is up to 11 times faster than the state of the art. Future work can examine computer graphics content more closely. We will release our datasets and code to encourage future research.

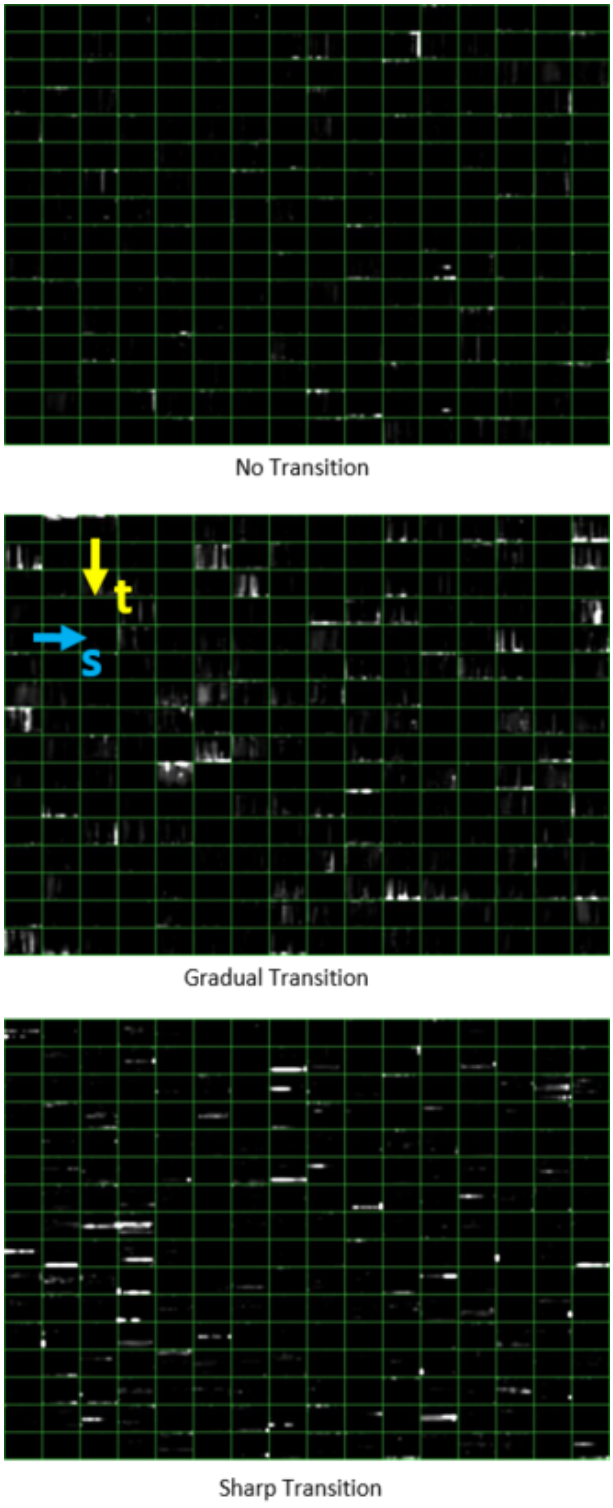

Fig. 6: Filter responses of DeepSBD stacked next to each other. The red grid shows some filters' borders. Here, y-axis is time and x-axis is space. Sharp transitions have an abrupt response in time (bright horizontal lines). Gradual transitions have blurred responses in time. No transition do not show specific patterns.

References

[1] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," *Computer Vision and Image Understanding (CVIU)*, vol. 114, no. 4, pp. 411–418, 2010.

[2] Z. Liu, E. Zavesky, D. Gibbson, B. Shahraray, and P. Haffner, "At&t research at trecvid 2007," in *TRECVID Workshop*, 2007.

[3] L. Priya and D. S., "Walsh hadamard transform kernel-based feature vector for shot boundary detection," *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 12, pp. 5187–5197, 2014.

[4] Z.-M. Lu and Y. Shi, "Fast video shot boundary detection based on svd and pattern matching," *TIP*, vol. 22, no. 12, pp. 5136–5145, 2013.

[5] P. P. Mohanta, S. K. Saha, and B. Chanda, "A model-based shot boundary detection technique using frame transition parameters," *IEEE Transactions on Multimedia (TMM)*, vol. 14, no. 1, pp. 223–233, 2012.

[6] D. Adjeroh, M. C. Lee, N. Banda, and U. Kandaswamy, "Adaptive edge-oriented shot boundary detection," *EURASIP Journal on Image and Video Processing*, vol. 2009, no. 1, 2009.

[7] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 16, no. 1, pp. 82–91, 2006.

[8] J. Lankinen and J.-K. Kämäräinen, "Video shot boundary detection using visual bag-of-words," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2013.

[9] D. Lelescu and D. Schonfeld, "Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 106–117, 2003.

[10] C. Zhang and W. Wang, "A robust and efficient shot boundary detection approach based on fisher criterion," in *ACM Multimedia*, 2012, pp. 701–704.

[11] D. M. Thounaojam, T. Khelchandra, K. M. Singh, and S. Roy, "A genetic algorithm and fuzzy logic approach for video shot boundary detection," *Computational intelligence and neuroscience*, vol. 2016, 2016.

[12] J. Yuan, W. Zheng, L. Ding, D. Wang, Z. Tong, H. Wang, J. L. J. Wu, F. Lin, and B. Zhang, "Tsinghua university at trecvid 2004: Shot boundary detection and high-level feature extraction," in *TRECVID Workshop*, 2004.

[13] M. W. Tao, J. Bai, P. Kohli, and S. Paris, "Simpleflow: A non-iterative, sublinear optical flow algorithm," *Computer Graphics Forum (Eurographics)*, vol. 31, no. 2, 2012.

[14] A. Dosovitskiy, P. Fischery, E. Ilg, P. Husser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*, 2015, pp. 2758–2766.

[15] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision (IJCV)*, vol. 92, no. 1, pp. 1–31, 2011.

[16] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015, pp. 4489–4497.

[17] L. Baraldi, C. Grana, and R. Cucchiara, "A deep siamese network for scene detection in broadcast videos," in *ACM Multimedia*, 2015, pp. 1199–1202.

[18] E. Apostolidis and V. Mezaris, "Fast shot segmentation combining global and local visual descriptors," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6583–6587.

[19] L. Baraldi, C. Grana, and R. Cucchiara, "Shot and scene detection via hierarchical clustering for re-using broadcast video," in *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2015, pp. 1–11.

[20] S. Lian, "Automatic video temporal segmentation based on multiple features," *Soft Computing*, vol. 15, no. 3, pp. 469–482, 2011.

[21] Y. Kawai, H. Sumiyoshi, and N. Yagi, "Shot boundary detection at trecvid 2007," in *TRECVID Workshop*, 2007.

[22] J. Yuan, H. Wang, L. Xiao, D. Wang, D. Ding, Y. Zuo, Z. Tong, X. Liu, S. Xu, W. Zheng, X. Li, Z. Si, J. Li, F. Lin, and B. Zhang, "Tsinghua university at trecvid 2005," in *TRECVID Workshop*, 2005.

[23] N. I. of Standards and Technology, "http://trecvid.nist.gov/trecvid.data.html," https://www.nist.gov/, 2017.

[24] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, pp. 1647–1654, 2007.

[25] R. T. Network, "The rai scuola video archives," http://www.scuola.rai.it/, 2015.

[26] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1212.html#abs-1212-0402